Amendments to the Drawings:

The attached sheet of drawings includes changes to FIG. 2. The attached sheet, which is labeled 'Replacement Sheet,' replaces the original sheet including FIG. 2. No new matter has been added.

REMARKS

The applicants have carefully considered the official action of June 11, 2008. Claims 1, 11, 13, 17, 19, 24, and 28 have been amended. In view of the foregoing amendments and the following remarks, the applicants respectfully request reconsideration of this application.

Amendments to Specification

By way of this response, the specification has been amended to correct typographical errors and to spell out a plurality of acronyms at their first appearance. It is respectfully submitted that new matter has been added.

Amendments to the Drawings

The applicants note that a replacement figure is submitted with this paper. The replacement figure, which replaces FIG. 2, is submitted to correct typographical errors related to the numbering of certain elements in the originally submitted figure. It is respectfully submitted that no new matter has been added and that every element of the replacement figure appears in the original figure, only with different reference numerals for certain elements.

Claim Objections

The applicants note that the examiner objected to claim 10, asserting that the acronym "ARM" should be spelled out at its first appearance. However, as shown in the amendment to paragraph [0007] above, the term "ARM" refers to a specific processor family and the associated instruction set. In particular, ARM (previously, the Advanced RISC (Reduced Instruction Set Computer), and prior to that the Acorn RISC Machine) is currently not an acronym, but rather a stand alone term.

35 U.S.C. §103 Rejections

Independent claims 1, 17, and 28 were rejected under 35 U.S.C. §103(a) as unpatentable over Goff (U.S. Patent No. 6,684,390) in view of Van Dyke et al. (U.S. Patent No. 7,013,456). As amended, claims 1, 17, and 28 recite a device or method of compiling a non-native software instruction to generate a first native software instruction from a first instruction set and, if the first native software instruction is executing more than a threshold

number of times, compiling the non-native software instruction to generate a second native software instruction from a second instruction set.

In contrast, Van Dyke et al. describe a first native code being translated into a second native code that is functionally equivalent to the first native code. Specifically, the system described by Van Dyke et al. converts the first native code (which is based on a first instruction set, such as X86) into the second native code (which is based on a second instruction set, such as Tapestry) by mapping resources related to the first native code to resources related to the second native code. For example, Van Dyke et al. employ a converter that decodes an X86 instruction and decomposes it into one or more Tapestry instructions that Van Dyke et al. refer to as the "recipe" for the corresponding X86 instruction. Column 15, lines 17-20.

The Van Dyke et al. technique is confirmed in the Office action by the examiner, who states that the "Tapestry/TAXi native instructions [are] based on native X86 instructions." See, *the office action dated June 11, 2008, page 4.* Thus, Van Dyke et al. do not describe compiling a non-native software instruction to generate the second native instruction, but translates one native instruction into another native instruction in certain instances. Accordingly, Van Dyke et al. do not describe the recitations of claim 1, 17, or 28.

Moreover, Goff does not cure the deficiencies of Van Dyke et al. Goff describes a system in which the processing of non-JAVA applications occurs at a host processor and the processing of JAVA applications occurs at an auxiliary system to distribute the work load. See, column 4, lines 47-54. However, Goff does not describe a device or method of compiling a non-native software instruction to generate a first native software instruction from a first instruction set and, if the first native software instruction is executing more than a threshold number of times, compiling the non-native software instruction to generate a second native software instruction from a second instruction set. For at least this reason, no combination of Goff and Van Dyke et al. describes all of the recitations of independent claims 1, 17, or 28 and, thus, the obviousness rejections thereof must be withdrawn.

Further, to support an obviousness rejection, the examiner bears the initial burden of factually supporting a *prima facie* conclusion of obviousness. See, *M.P.E.P. 2142.* Further, as recognized by the Supreme Court of the United States, the analysis supporting a rejection under 35 U.S.C. §103 must be made explicit. *In re Kahn*, 441 F. 3d 977, 988 (CA Fed. 2006) cited with approval in KSR ("[R]ejections on obviousness grounds cannot be sustained by

mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness."). Here, no motivation to combine or rationale has even been asserted, let alone factually supported.

In the official action, the examiner states that "[i]t would have been obvious to a person having ordinary skill in the art...to combine Van Dyke's teaching into Goff's teaching." See, *the official action dated June 11, 2008, page 4.* The examiner goes on to state that "one would have been motivated to do so to provide a multi-processor Java subsystem as suggested by Goff...and also [to] provide two instruction set architectures, which can use a hot spot detector to translate and optimize binary execution as suggested by Van Dyke." See, *the official action dated June 11, 2008, page 4.* Nothing further in support of the *prima facie* case of obviousness appears in the official action.

Given the lack of any further explanation or rationale, this statement does not establish a *prima facie* case of obviousness. The examiner has only described Goff and Van Dyke et al., separately, and summarily concluded that it would be beneficial to have the system described by Goff and the system described by Van Dyke et al. In fact, there is no suggestion in the references themselves or by the examiner that the two systems (i.e., the multi-processor subsystem described by Goff and the system described by Van Dyke et al.) would benefit in any way from each other. In other words, the examiner does not provide any type of potential interaction between the prior art systems or any reason to modify one of the systems with the other.

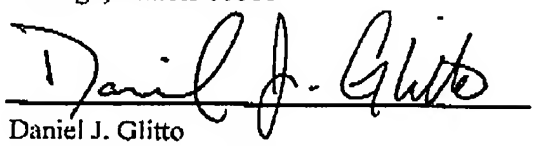
Because the official action fails to provide evidence, reasoning, or an articulation for the conclusory statements quoted above, the §103(a) rejections must be withdrawn.

The Commissioner is hereby authorized to charge any deficiency in the amount enclosed (if any) or any additional fees which may be required during the pendency of this application to Deposit Account No. 50-2455.

Respectfully submitted,

HANLEY, FLIGHT & ZIMMERMAN, LLC
150 South Wacker Drive
Suite 2100
Chicago, Illinois 60606

August 11, 2008



Daniel J. Glitto
Registration No. 58,996
Attorney for Applicant(s)